

REMARKS

Claims 13 and 33 have been canceled. Claims 1, 2, 4, 14, 18, 19, 23, 24, 31, 37, 42, 45, 58, 59, 63, 64, and 69-92 have been amended. Claims 1-12, 14-32 and 34-92 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Double Patenting Rejection:

The Office Action provisionally rejected claims 1, 3, 8-10, 14, 17, 45, 47, 52-54, 56, 57, 69, 71, 76-78, 80 and 81 under the judiciary created doctrine of obviousness-type double patenting as being unpatentable over claims 20, 23, 26-28, 35, 37, 38, 41, 44-46, 53 and 55 of co-pending Application No. 10/642,928. Should this rejection become non-provisional, Applicants will consider filing a terminal disclaimer or present reasons traversing the rejection.

Section 101 Rejection:

The Office Action rejected claims 45-92 under 35 U.S.C. § 101 because the claimed invention is allegedly directed to non-statutory subject matter.

In regard to claims 45-68, the Office Action asserts that “no other statutory classes is recited as all of the steps recited as part of the method...can be performed by a person using a paper and pencil,” and that, in addition, “no physical transformation is recited.” Applicant respectfully traverse this rejection for at least the reason that “implementing the Web Service integrated with the business system according to the integrated Web Service architecture” clearly recites a transformation of the business system. However, to expedite prosecution, claim 45 has been amended to recite “generating, by an integrated Web Services architecture design mechanism implemented on one or more computers, an integrated Web Service architecture comprising a plurality of heterogeneous components of the business system in accordance with one or more

integration design patterns.”

Thus, Applicants respectfully request removal of the § 101 rejection of claims 45-68.

In regard to claims 69-92, the Office Action rejects these claims under § 101 because the “computer-accessible medium” may be a transmission media or communications media according to Applicant’s specification, page 248, first paragraph. Applicant respectfully traverses this rejection. However, to expedite prosecution, claims 69-92 have been amended to recite “A computer-accessible storage medium storing program instructions, wherein the program instructions are computer-executable to implement...”

Thus, Applicants respectfully request removal of the § 101 rejection of claims 69-92.

Section 103(a) Rejection:

The Office Action rejected claims 1-3, 5, 6, 8-10, 13-24, 27-31, 33-35, 37-39, 41-43, 45-47, 49, 50, 52-54, 56-64, 67-71, 73, 74, 76-78, 80-88, 91 and 92 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry et al. (U.S. Publication 2003/0233631) (hereinafter “Curry”) in view of Huang et al. (“A Web Services-Based Framework for Business Integration Solutions”) (hereinafter “Huang”). Applicant respectfully traverses this rejection for at least the following reasons.

In regard to claim 1, as a first matter, neither of the cited references (Curry and Huang) teach program instructions executable to implement a Web Services architecture design service for generating integrated Web Service architectures for integrating Web Services with business systems.

Curry teaches a “Web services development method” (Title). Curry’s Abstract describes “A method for rapid design, development, deployment and support of web applications based on web services with minimum customized programming, maximized reuse of software components and compliance with standard development frameworks.” Curry’s FIG. 1 illustrates “a flow chart illustrating the steps of a rapid web services development method according to an illustrative embodiment of the present invention.” (paragraph [0030]). However, nowhere does Curry teach that the method illustrated in FIG. 1 is a computer-implemented method, and Curry does not describe a computer system that implements the method illustrated in FIG. 1. At most, Curry describes that particular steps or sub-steps of Curry’s method may be assisted by or performed using existing software tools. For example, in paragraph [0072], cited by the Office, Curry states: “In an illustrative embodiment of the invention, the Epiowave™ environment available from the applicant, Epionet Corporation of Dublin, Ireland, serves as the D&D environment in the D&D application creation step 32.” As another example, in paragraph [0076], also cited by the Office, Curry states (emphasis added): “In an illustrative embodiment of the invention, a toolset called MindManager by MindJet LLC of Sausalito, Calif. is used to create mind-maps which assist in the steps of creating the Role Control Diagram 44 and performing the Use Case Analysis 46.”

Huang teaches a “Web services-based framework for business integration solutions” (Title). In Section 3, at page 18, second column, first full paragraph, Huang teaches “In this section, we propose a framework to develop Web services-based business integration solutions.” However, Huang only conceptually describes this framework, and does not teach or even suggest a computer implementation of the framework for developing Web services-based business integration solutions.

In contrast to both Curry and Huang, claim 1 teaches a system for integrating Web Services with business systems, comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to generate integrated Web Service architectures for integrating Web Services with business systems, wherein, to generate an integrated Web Service architecture for integrating a

specific Web Service with a specific business system, the program instructions are executable by the processor to: generate the integrated Web Service architecture comprising a plurality of heterogeneous components of the specific business system in accordance with one or more integration design patterns...and provide output indicating the generated integrated Web Service architecture for integrating the specific Web Service with the specific business system.

Furthermore, at paragraph [0022], Curry teaches (emphasis added): “Interface templates are also reviewed for compliance with a standard framework architecture.” In paragraph [0055], Curry teaches (emphasis added): “The framework as referred to in the present invention is an overall architecture which provides a template for building enterprise web solutions. The framework includes a pre-built architecture that allows developers to rapidly create applications based on business components and web services.” Thus, **Curry does not teach a system comprising program instructions executable to implement a Web Services architecture design service configured to generate an integrated Web Service architecture for integrating a specific Web Service with a specific business system.** Instead, Curry clearly teaches a “standard framework architecture” that clearly pre-exists and that may allegedly be used to “rapidly create applications based on business components and web services.”

Neither Curry nor Huang, alone or in combination, teach a system as recited in Applicant’s claim 1 when the claim is viewed as a whole.

In further regard to claim 1, the Office has failed to establish that the cited references teach the features of, “A system for integrating Web Services with business systems, comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to generate integrated Web Service architectures for integrating Web Services with business systems.”

The Office Action cites Curry, paragraph [0072] and paragraph [0076], as teaching a system for integrating Web Services with business systems, comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor. However, neither of these citations actually mentions a system comprising a processor and a memory comprising program instructions. As previously noted, these citations from Curry only teach that particular steps or sub-steps of Curry's method as illustrated in FIG. 1 may be assisted by or performed using existing software tools. Huang is also silent in regards to a system comprising a processor and memory comprising program instructions as recited in Applicant's claim 1.

Moreover, as discussed above, neither Curry nor Huang, alone or in combination, teach a system for integrating Web Services with business systems, comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to generate integrated Web Service architectures for integrating Web Services with business systems. Curry does not describe a computer system that implements the method illustrated in FIG. 1. Huang only conceptually describes a framework, and does not teach or even suggest a computer implementation of the framework.

In further regard to claim 1, the cited references fail to teach at least the features of program instructions executable by a processor to generate an integrated Web Service architecture [for integrating a specific Web Service with a specific business system], where the integrated Web Service architecture comprises a plurality of heterogeneous components of the specific business system, and provide output indicating the generated integrated Web Service architecture for integrating the specific Web Service with the specific business system.

The Office Action cites Curry, paragraphs [0021], [0022], and [0024] as teaching program instructions executable by a processor to generate an integrated Web Service architecture comprising a plurality of heterogeneous components of a business system.

Paragraph [0021] broadly summarizes Curry's method and alleged advantages thereof, and does not teach program instructions executable by a processor to generate an integrated Web Service architecture as recited in claim 1.

Paragraph [0022] broadly summarizes Curry's method as illustrated in Curry's FIG. 1. The paragraph states that business logic development steps are used to "efficiently collect and refine the various business requirements of a customer into a set of standard specification documents," and that Web service mapping steps are then used to "organize the business requirements into a set of prioritized web services." However, this does not teach program instructions executable by a processor to generate an integrated Web Service architecture as recited in claim 1.

Paragraph [0022] of Curry goes on to teach "Interface templates are also reviewed for compliance with a standard framework architecture." However, in paragraph [0055], Curry teaches (emphasis added): "The framework as referred to in the present invention is an overall architecture which provides a template for building enterprise web solutions. The framework includes a pre-built architecture that allows developers to rapidly create applications based on business components and web services." Thus, **Curry does not teach a method for generating this "standard framework architecture."** Instead, Curry clearly teaches that this "standard framework architecture" pre-exists and allegedly may be used to "rapidly create applications based on business components and web services."

Paragraph [0022] of Curry goes on to teach "The standard framework includes a Function Implementation System which automates the creation of web service business functions." In paragraph [0024], Curry further teaches "The function implementation system which uses a standard application template and an application generator to automatically create business objects is continuously enhanced by the addition of new business objects during the application creation steps." Curry's "function implementation system" thus clearly is used to create "web service business functions"

using a “standard application template” and an “application generator” during the “application creation steps” of Curry’s method as illustrated in FIG. 1. However, it is clear that Curry’s “function implementation system” does not teach or suggest program instructions executable by a processor to generate an integrated Web Service architecture as recited in claim 1.

The rest of Curry’s paragraph [0024] teaches that “Classes and components are designed according to a best practices business object framework. Web service applications are created by integrating the classes and components according to the business object framework.” This does not teach or suggest program instructions executable by a processor to generate an integrated Web Service architecture as recited in claim 1.

Furthermore, Applicant’s claim 1 goes on to recite that the program instructions are executable by a processor to provide output indicating the generated integrated Web Service architecture for integrating the specific Web Service with the specific business system. Nowhere in these citations or elsewhere does Curry teach providing output indicating a generated integrated Web Service architecture for integrating a specific Web Service with a specific business system as recited in claim 1.

Moreover, as discussed above, Huang only conceptually describes a framework, and does not teach or even suggest a computer implementation of the framework, nor does Huang teach or suggest program instructions executable by a processor to: generate an integrated Web Service architecture, where the integrated Web Service architecture comprises a plurality of heterogeneous components of the specific business system; and provide output indicating the generated integrated Web Service architecture for integrating the specific Web Service with the specific business system. Moreover, no combination of Huang and Curry teaches these features as recited in Applicant’s claim 1.

In further regard to claim 1, the Office has failed to establish that the cited references teach program instructions executable by a processor to generate one or

more Use Cases for the integrated Web Service.

The Office Action relies on Curry to teach these features, citing Curry, paragraphs [0080]-[0084]. These paragraphs describe a step 46 in Curry's FIG. 2, which expands on element 10 ("Business Logic Development") of Curry's FIG. 1 (see paragraph [0074]). At paragraph [0074], Curry states "During the business logic development step 10, a precise description of the customer's business requirements or the business logic requirements of the web services under development are compiled by performing an orderly sequence of steps." As previously noted, Curry does not teach that the method illustrated in FIG. 1 is a computer-implemented method, and Curry does not describe a computer system that implements the method illustrated in FIG. 1. In paragraph [0076], Curry states (emphasis added): "First, a set of Role Control Diagrams 44 is developed and a Use Case Analysis is performed 46. In an illustrative embodiment of the invention, a toolset called MindManager by MindJet LLC of Sausalito, Calif. is used to create mind-maps which assist in the steps of creating the Role Control Diagram 44 and performing the Use Case Analysis 46." In paragraph [0080], Curry states "Next, a Use Case Analysis step 46 is performed during which the functions attached to each role are broken down into a set of use cases." Thus, Curry only teaches the use of a "toolset" to "assist" in the step of performing a Use Case Analysis, and that during the Use Case Analysis "the functions attached to each role are broken down into a set of use cases." In contrast, Applicant's claim 1 specifically recites program instructions executable by a processor to generate one or more Use Cases for the integrated Web Service.

In further regard to claim 1, the Office has failed to establish that the cited references teach program instructions executable by a processor to generate a high-level architecture for the integrated Web Service, wherein the high-level architecture identifies two or more entities of the integrated Web Service and the relationships and interactions among the entities.

The Office Action relies on Curry to teach these features, citing paragraph [0097] and asserting "EN: the context diagram describes the high-level architecture." Paragraph

[0097] states (emphasis added): “Once the various forms of project descriptions, including business rules 52, state diagrams 54, swim-lane diagrams 56 and activity diagrams 58 are complete, a context diagram 60 is prepared.” Applicant notes that the context diagram 60 is prepared at step 10 (“Business Logic Development”) of Curry’s method as illustrated in FIG. 1 (see Curry’s FIG. 2, which provides details of step 10). As previously noted, Curry does not teach that the method illustrated in FIG. 1 is a computer-implemented method, and Curry does not describe a computer system that implements the method illustrated in FIG. 1. While Curry teaches that a “context diagram 60 is prepared,” Curry does not teach program instructions executable by a processor to generate a context diagram 60. In contrast, claim 1 specifically recites program instructions executable by a processor to generate a high-level architecture for the integrated Web Service

In further regard to claim 1, the Office has failed to establish that the cited references teach program instructions executable by a processor to generate a logical architecture for the integrated Web Service according to the Use Cases, wherein the logical architecture identifies two or more logical components of the integrated Web Service and the relationship among the logical components, and wherein the logical architecture comprises two or more layers.

The Office Action relies on Curry to teach these features, citing paragraph [0055]-[0059] and asserting “EN: the framework structure including a number of layers is the logical architecture.” In paragraph [0055], Curry teaches (emphasis added): “The framework as referred to in the present invention is an overall architecture which provides a template for building enterprise web solutions. The framework includes a pre-built architecture that allows developers to rapidly create applications based on business components and web services.” Thus, **Curry does not teach generating this “standard framework architecture.”** More specifically Curry does not teach program instructions executable by a processor to generate this “standard framework architecture.” Instead, Curry only teaches that this “standard framework architecture” pre-exists and allegedly may be used to “rapidly create applications based on business components and web

services.”

Furthermore, paragraph [0055] states that “A framework review 18 can then be performed on the interface templates.” Paragraph [0060] of Curry states (emphasis added): “During the framework review 18 of the interface templates, a developer reviews the prototype from a framework perspective and insures that the interface templates are achievable in a framework context.” Thus, step 18 (the framework review step), described in paragraphs [0055]-[0059], is performed on Curry’s “interface templates.” Moreover, the framework review step 18 is performed by a human (“developer”) and not by “program instructions executable by a processor.”

In further regard to claim 1, the Office has failed to establish that the cited references teach program instructions executable by a processor to generate an integrated Web Service architecture comprising a plurality of heterogeneous components of a specific business system in accordance with one or more integration design patterns.

The Office Action admits that Curry does not teach generating an integrated Web Service architecture in accordance with one or more integration design patterns, and relies on Huang to teach these features, citing page 18, second column, paragraph 2 and page 20, first column paragraphs 2 and 3 and second column, paragraphs 1-3. In Section 3, at page 18, second column, first full paragraph, Huang teaches “In this section, we propose a framework to develop Web services-based business integration solutions.” In the first paragraph cited from page 20, Huang teaches “In essence, this framework uses four different design patterns...” However, Huang only conceptually describes this framework, and does not teach or even suggest a computer implementation of the framework for developing Web services-based business integration solutions. As noted above, Curry also fails to teach a computer implementation of Curry’s method into which Huang’s “design patterns” might be combined. The proposed combination of Huang with Curry would not result in program instructions executable by a processor to generate an integrated Web Service architecture comprising a plurality of heterogeneous

components of a specific business system in accordance with one or more integration design patterns, as recited in Applicant's claim 1.

In further regard to claim 1, the Office has failed to establish a proper *prima facie* reason to combine the references.

The Office Action asserts that "it would have been obvious...to have modified Curry such that the web service architecture is generated in accordance with one or more integration design patterns as taught by Huang because design patterns are well known in the art and commonly used by programmers since design patterns provide the benefit of capturing a standard solution to a common programming problem for reuse." However, Curry teaches four specific design patterns – a Composite pattern, a Mediator pattern, a Command pattern, and a State pattern. It is unclear as to how Curry's system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry's "Web services development method" as disclosed. Curry and Huang each propose distinctly different methods for achieving similar results. Modifying Curry with Huang's specific "design patterns" would appear to necessitate significant changes in Curry's system. "If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious." *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). Thus, one of ordinary skill would not have combined the teachings of Huang with the teachings of Curry in the manner proposed by the Office. Accordingly, the Office has failed to establish a *prima facie* case of obviousness.

Thus, for at least the reasons presented above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 1 also apply to claims 45 and 69.

In regard to claim 18, as a first matter, neither of the cited references (Curry and Huang) teach program instructions executable to implement a Web Services architecture design service for generating integrated Web Service architectures.

As noted above in regards to claim 1, nowhere does Curry teach that the method illustrated in FIG. 1 is a computer-implemented method, and Curry does not describe a computer system that implements the method illustrated in FIG. 1. At most, Curry describes that particular steps or sub-steps of Curry's method may be assisted by or performed using existing software tools. Huang only conceptually describes a framework for developing Web services-based business integration solutions, and does not teach or even suggest a computer implementation of the framework for developing Web services-based business integration solutions.

In contrast to both Curry and Huang, claim 18 teaches a system for generating integrated Web Service architectures, comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to generate integrated Web Service architectures for implementing integrated Web Service business systems, wherein, to generate an integrated Web Service architecture for implementing a specific integrated Web Service business system, the program instructions are executable by the processor to: identify one or more components of the integrated Web Service architecture according to one or more use case requirements for the specific integrated Web Service business system; define a plurality of integration tiers and one or more Web Services technologies according to a Web Services architecture integration framework; define how each of the plurality of integration tiers communicates with others of the plurality of integration tiers according to the Web Services architecture integration framework; organize the components according to the plurality of integration tiers and two or more layers of the integrated Web Service architecture; apply one or more design patterns to the integrated Web Service architecture; and provide output indicating the generated integrated Web Service architecture for implementing the specific integrated Web Service business system.

Furthermore, as noted above in regard to claim 1, in paragraph [0055] Curry teaches “The framework includes a pre-built architecture that allows developers to rapidly create applications based on business components and web services.” **Curry does not teach a system comprising program instructions executable to implement a Web Services architecture design service configured to generate an integrated Web Service architecture for implementing a specific integrated Web Service business system.** Instead, Curry clearly teaches a “standard framework architecture” that clearly pre-exists and that may allegedly be used to “rapidly create applications based on business components and web services.”

Neither Curry nor Huang, alone or in combination, teach a system as recited in Applicant’s claim 18 when the claim is viewed as a whole.

In further regard to claim 18, the Office has failed to establish that the cited references teach, “A system for generating integrated Web Service architectures, comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to implement a Web Services architecture design service configured to generate integrated Web Service architectures for implementing integrated Web Service business systems.”

The Office Action cites Curry, paragraph [0072] and paragraph [0076], as teaching a system for generating integrated Web Service architectures, comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor. However, neither of these citations actually mentions a system comprising a processor and a memory comprising program instructions. As previously noted, these citations from Curry only teach that particular steps or sub-steps of Curry’s method as illustrated in FIG. 1 may be assisted by or performed using existing software tools. Huang is also silent in regards to a system comprising a processor and memory comprising program instructions as recited in Applicant’s claim 18.

Moreover, as previously discussed, neither Curry nor Huang, alone or in combination, teach a system for generating integrated Web Service architectures, comprising: a processor; and a memory comprising program instructions, wherein the program instructions are executable by the processor to generate integrated Web Service architectures for integrating Web Services with business systems. Curry does not describe a computer system that implements the method illustrated in FIG. 1. Huang only conceptually describes a framework, and does not teach or even suggest a computer implementation of the framework.

In further regard to claim 18, the Office has failed to establish that the cited references teach, program instructions executable by a processor to: identify one or more logical components of the integrated Web Service architecture according to one or more use case requirements for the specific integrated Web Service business system.

The Office Action relies on Curry to teach these features, citing FIG. 2, paragraphs [0051]-[0052], and paragraphs [0074]-[0080]. FIG. 2 expands on element 10 (“Business Logic Development”) of Curry’s FIG. 1 (see paragraph [0074]). As previously noted, Curry does not teach that the methods illustrated in FIG. 1 and FIG. 2 are computer-implemented methods, and Curry does not describe a computer system that implements the methods illustrated in FIG. 1 and FIG. 2. In paragraph [0076], Curry states (emphasis added): “First, a set of Role Control Diagrams 44 is developed and a Use Case Analysis is performed 46. In an illustrative embodiment of the invention, a toolset called MindManager by MindJet LLC of Sausalito, Calif. is used to create mind-maps which assist in the steps of creating the Role Control Diagram 44 and performing the Use Case Analysis 46.” In paragraph [0080], Curry states “Next, a Use Case Analysis step 46 is performed during which the functions attached to each role are broken down into a set of use cases.” Thus, Curry only teaches the use of a “toolset” to “assist” in the step of performing a Use Case Analysis, and that during the Use Case Analysis “the functions attached to each role are broken down into a set of use cases.”

In paragraphs [0084]-[0085], Curry teaches “Each of the use cases is defined by breaking it down to a sufficient level of granularity. Completion of the use case analysis 46 results in a detailed picture of the interrelationships in the information environment of the web service under development. Once a complete set of documents and/or diagrams is developed according to the business logic development step 10, the use case analysis 46 is complete. According to the illustrative embodiment of the invention, these documents include any of or any combination of business rules 52, state diagrams 54, swimlane diagrams 56, and/or activity diagrams 58.” At paragraph [0097], Curry goes on to teach “Once the various forms of project descriptions, including business rules 52, state diagrams 54, swim-lane diagrams 56 and activity diagrams 58 are complete, a context diagram 60 is prepared.” The context diagram 60 is prepared at step 10 (“Business Logic Development”) of Curry’s method as illustrated in FIG. 1 (Curry’s FIG. 2 provides details of step 10). Curry does not teach that the methods illustrated in FIGs. 1 and 2 are computer-implemented methods, and Curry does not describe a computer system that implements the methods illustrated in FIGs. 1 and 2. While Curry teaches that a “context diagram 60 is prepared,” Curry does not teach program instructions executable by a processor to generate a context diagram 60.

Thus, Curry does not teach program instructions executable by the processor to identify one or more logical components of the integrated Web Service architecture according to one or more use case requirements for the specific integrated Web Service business system, as recited in Applicant’s claim 18. Huang, who only teaches a conceptual framework and like Curry fails to teach a computer implementation of the framework, fails to overcome these shortcomings of Curry.

In further regard to claim 18, the cited references do not teach at least the features of, program instructions executable by a processor to: translate the one or more use case requirements for the specific integrated Web Service business system and one or more technical constraints for the specific integrated Web Service business system to determine a plurality of Web Service components for the integrated Web Service architecture, wherein the Web Service components include

software components; [and] categorize the Web Service components into two or more related groups according to a Web Services architecture integration framework, as recited in amended claim 18.

Neither Curry nor Huang, alone or in combination, teach program instructions executable by a processor to translate one or more use case requirements for a specific integrated Web Service business system and one or more technical constraints for the specific integrated Web Service business system to determine a plurality of Web Service components for an integrated Web Service architecture. Furthermore, neither Curry nor Huang, alone or in combination, teach program instructions executable by a processor to categorize the Web Service components into two or more related groups according to a Web Services architecture integration framework.

In further regard to claim 18, the Office has failed to establish that the cited references teach, program instructions executable by a processor to: define a plurality of integration tiers for the integrated Web Service architecture and one or more Web Services technologies for the integrated Web Service architecture according to the Web Services architecture integration framework.

The Office Action relies on Curry to teach these features, citing paragraphs [0056]-[0059] and [0151], and asserting “logical layers of the application may be each physically separated or may be combined into components that include multiple logical layers.”

Paragraphs [0056]-[0059] are in reference to step 18 (“framework review”) of Curry’s FIG. 1. See paragraph [0055]: “A framework review 18 can then be performed on the interface templates. The framework as referred to in the present invention is an overall architecture which provides a template for building enterprise web solutions. The framework includes a pre-built architecture that allows developers to rapidly create applications based on business components and web services.” Paragraph [0056] goes on to state: “The framework structure reflects the essential make-up of a well built enterprise

solution. The structure includes a number of layers or sub-architectures.” However, **Curry does not teach program instructions executable by a processor to define this “standard framework architecture” or the layers or sub-architectures of the framework structure.** Instead, Curry only teaches that this “standard framework architecture” pre-exists and allegedly may be used to “rapidly create applications based on business components and web services.” Furthermore, paragraph [0060] of Curry states that “During the framework review 18 of the interface templates, a developer reviews the prototype from a framework perspective and insures that the interface templates are achievable in a framework context.” Thus, the framework review step is performed by a human “developer” and not by “program instructions executable by a processor.”

Paragraph [0151], on the other hand, is in reference to step 28 (“Class and Component Design”) of Curry’s FIG. 1. See paragraph [0148]: “Next, referring back to FIG. 1 according to an illustrative embodiment of the invention, a Class and Component Design step 28 is performed. The Class and Component Design step 28 is described in greater detail with reference to FIG. 15. This step is typically performed in strict compliance with the requirements of the Business Object Framework architecture.” In paragraph [0057], Curry identifies the Business Object Framework architecture as one of the “layers” or “sub-architectures” of the “framework structure”: “The framework includes primarily three main sub-architectures. One of the sub-architecture is referred to as the Business Object Framework (BOF). The BOF is basically a template for creating components which provide the main processing logic within an application.”

Paragraph [0151] goes on to state: “Once the classes have been created, a Review Structure step 166 is typically performed to review class and component structure for the application. At this point, according to the illustrative embodiment, decision is made as to how the created classes and components will be combined within the application. Several architectural options are available. For example, the logical layers of the application may be each physically separated or may be combined into components that include multiple logical layers.” However, **Curry does not teach program instructions executable by a**

processor to perform Class and Component Design step 28 or Review Structure step 166 of FIG. 15.

Furthermore, Curry states that “This step [28] is typically performed in strict compliance with the requirements of the Business Object Framework architecture.” Therefore, Review Structure step 166 is “typically performed in strict compliance with the requirements of the Business Object Framework architecture.” Since the Business Object Framework architecture is given as one of the “the primarily three main sub-architectures” of the framework structure described in paragraphs [0056]-[0059], it appears that Curry’s use of “layers” in paragraphs [0056]-[0059] as another term for the sub-architectures described therein is different than Curry’s use of “layers” in paragraph [0151]. In other words, the “layers” in paragraph [0151] refer to something distinctly different than “layers” as mentioned in paragraph [0056] (“The structure includes a number of layers or sub-architectures”).

From the above, the Office has improperly cited two distinct and different sections from Curry (paragraphs [0056]-[0059] related to step 18 (“framework review”) of Curry’s FIG. 1 and to Curry’s “framework structure”, and paragraph [0151], related to step 28 (“Class and Component Design”) of Curry’s FIG. 1 that is “is “typically performed in strict compliance with the requirements of the Business Object Framework architecture”), in an attempt to support the assertion that Curry teaches this subject matter as recited in claim 18.

In further regard to claim 18, the Office has failed to establish that the cited references teach, program instructions executable by a processor to: define how each of the plurality of integration tiers communicates with others of the plurality of integration tiers in the integrated Web Service architecture according to the Web Services architecture integration framework.

The Office Action relies on Curry to teach these features, citing paragraphs [0056], [0136], and [0151], and asserting “the sub-architectures are linked using XML as a standard communication mechanism.”

As previously noted, paragraph [0056] is in reference to step 18 (“framework review”) of Curry’s FIG. 1, as described in paragraph [0055]. Paragraph [0056] states: “The sub-architectures are linked using XML as a standard communication mechanism to provide an integrated, structured and extensible development environment.” However, **Curry does not teach program instructions executable by a processor to define how each of a plurality of integration tiers communicates with others of the plurality of integration tiers in the integrated Web Service architecture according to the Web Services architecture integration framework.** Instead, Curry only teaches that this “standard framework architecture” pre-exists and that “the sub-architectures are linked using XML as a standard communication mechanism.” Furthermore, paragraph [0060] of Curry states that “During the framework review 18 of the interface templates, a developer reviews the prototype from a framework perspective and insures that the interface templates are achievable in a framework context.” Thus, the framework review step is performed by a human “developer” and not by “program instructions executable by a processor.”

Paragraph [0136] is in reference to step 22 (“Architectural Analysis”) of Curry’s FIG. 1. See paragraph [0135] (emphasis added): “Next, referring again to FIG. 1, an Architectural Analysis step 22 is performed which will be described in greater detail with respect to FIG. 12. The Architectural Analysis step 22, according to an illustrative embodiment of the present invention is purely an intellectual exercise performed according to known best practices of the art and does not rely on any particular software tools.” Paragraph [0136] describes a “Logical System Architecture Analysis 126 step” of Architectural Analysis step 22. Thus, **Curry clearly does not teach program instructions executable by a processor to perform step 22 (“Architectural Analysis”) of Curry’s FIG. 1 or Logical System Architecture Analysis 126 step of FIG. 12.**

Paragraph [0151], as previously noted, is in reference to step 28 (“Class and Component Design”) of Curry’s FIG. 1, as described in paragraph [0148]. However, **Curry does not teach program instructions executable by a processor to perform Class and Component Design step 28 or Review Structure step 166 of FIG. 15.** Moreover, this citation does not mention anything similar to defining how each of a plurality of integration tiers communicates with others of the plurality of integration tiers in an integrated Web Service architecture according to a Web Services architecture integration framework.

Moreover, the Office has improperly cited three distinct and different sections from Curry (paragraph [0056] related to step 18 (“framework review”) of Curry’s FIG. 1 and to Curry’s “framework structure”, paragraph [0136] related to step 22 (“Architectural Analysis”) of Curry’s FIG. 1, which Curry describes as “purely an intellectual exercise,” and paragraph [0151], related to step 28 (“Class and Component Design”) of Curry’s FIG. 1 that is “is “typically performed in strict compliance with the requirements of the Business Object Framework architecture”), in an attempt to support the assertion that Curry teaches this subject matter as recited in claim 18.

In further regard to claim 18, the cited references fail to teach, program instructions executable by a processor to: organize the groups of Web Service components according to the plurality of integration tiers and two or more layers of the integrated Web Service architecture, as recited in amended claim 18.

The Office Action relies on Curry to teach these features, citing Curry, paragraphs [0057]-[0059] and [0151]. Applicant first notes that the Office relied on Curry, paragraphs [0056]-[0059] and [0051], to allegedly teach “program instructions executable by a processor to: define a plurality of integration tiers for the integrated Web Service architecture and one or more Web Services technologies for the integrated Web Service architecture according to the Web Services architecture integration framework.” Thus, the Office is improperly relying on the same teaching from Curry to teach two distinct and different elements of Applicant’s claim 18. Moreover, this subject matter recites

“program instructions executable by a processor to: organize the groups of Web Service components according to the plurality of integration tiers and two or more layers of the integrated Web Service architecture.” Since the Office relies on the same teachings from Curry to teach “defining a plurality of integration tiers for the integrated Web Service architecture,” it is unclear as to what in these citations is supposed to be referring to the “integration tiers” and what to the “two or more layers of the integrated Web Service architecture” as recited in the claims.

Furthermore, claim 18 recites that the Web Service components are categorized into two or more related groups according to a Web Services architecture integration framework, and that the groups of Web Service components are then organized according to the plurality of integration tiers and two or more layers of the integrated Web Service architecture. Applicant can find nothing in Curry that teaches organizing groups of Web Service components according to a plurality of integration tiers and two or more layers of an integrated Web Service architecture.

Furthermore, paragraphs [0057]-[0059] are in reference to step 18 (“framework review”) of Curry’s FIG. 1. See paragraph [0055]: “A framework review 18 can then be performed on the interface templates. The framework as referred to in the present invention is an overall architecture which provides a template for building enterprise web solutions. The framework includes a pre-built architecture that allows developers to rapidly create applications based on business components and web services.” Paragraph [0056] goes on to state: “The framework structure reflects the essential make-up of a well built enterprise solution. The structure includes a number of layers or sub-architectures.” However, **Curry does not teach program instructions executable by a processor to define this “standard framework architecture” or the layers or sub-architectures of the framework structure as described in paragraphs [0057]-[0059].** Instead, Curry only teaches that this “standard framework architecture” pre-exists and allegedly may be used to “rapidly create applications based on business components and web services.” Furthermore, paragraph [0060] of Curry states that “During the framework review 18 of the interface templates, a developer reviews the prototype from a framework perspective

and insures that the interface templates are achievable in a framework context.” Thus, the framework review step is performed by a human “developer” and not by “program instructions executable by a processor.”

Paragraph [0151], on the other hand, is in reference to step 28 (“Class and Component Design”) of Curry’s FIG. 1. See paragraph [0148]: “Next, referring back to FIG. 1 according to an illustrative embodiment of the invention, a Class and Component Design step 28 is performed. The Class and Component Design step 28 is described in greater detail with reference to FIG. 15. This step is typically performed in strict compliance with the requirements of the Business Object Framework architecture.” In paragraph [0057], Curry identifies the Business Object Framework architecture as one of the “layers” or “sub-architectures” of the “framework structure”: “The framework includes primarily three main sub-architectures. One of the sub-architecture is referred to as the Business Object Framework (BOF). The BOF is basically a template for creating components which provide the main processing logic within an application.”

Paragraph [0151] goes on to state: “Once the classes have been created, a Review Structure step 166 is typically performed to review class and component structure for the application. At this point, according to the illustrative embodiment, decision is made as to how the created classes and components will be combined within the application. Several architectural options are available. For example, the logical layers of the application may be each physically separated or may be combined into components that include multiple logical layers.” However, **Curry does not teach program instructions executable by a processor to perform Class and Component Design step 28 or Review Structure step 166 of FIG. 15.**

Furthermore, Curry states that “This step [28] is typically performed in strict compliance with the requirements of the Business Object Framework architecture.” Therefore, Review Structure step 166 is “typically performed in strict compliance with the requirements of the Business Object Framework architecture.” Since the Business Object Framework architecture is given as one of the “the primarily three main sub-

architectures” of the framework structure described in paragraphs [0056]-[0059], it appears that Curry’s use of “layers” in paragraphs [0056]-[0059] as another term for the sub-architectures described therein is different than Curry’s use of “layers” in paragraph [0151]. In other words, the “layers” in paragraph [0151] refer to something distinctly different than “layers” as mentioned in paragraph [0056] (“The structure includes a number of layers or sub-architectures”).

From the above, the Office has improperly cited two distinct and different sections from Curry (paragraphs [0057]-[0059] related to step 18 (“framework review”) of Curry’s FIG. 1 and to Curry’s “framework structure”, and paragraph [0151], related to step 28 (“Class and Component Design”) of Curry’s FIG. 1 that is “is “typically performed in strict compliance with the requirements of the Business Object Framework architecture”), in an attempt to support the assertion that Curry teaches this subject matter as recited in claim 18.

In further regard to claim 18, the Office has failed to establish that the cited references teach program instructions executable by a processor to apply one or more design patterns to the integrated Web Service architecture.

The Office Action admits that Curry does not teach applying one or more design patterns to the integrated Web Service architecture, and relies on Huang to teach these features, citing page 18, second column, paragraph 2 and page 20, first column paragraphs 2 and 3 and second column, paragraphs 1-3. In Section 3, at page 18, second column, first full paragraph, Huang teaches “In this section, we propose a framework to develop Web services-based business integration solutions.” In the first paragraph cited from page 20, Huang teaches “In essence, this framework uses four different design patterns...” However, Huang only conceptually describes this framework, and does not teach or even suggest a computer implementation of the framework for developing Web services-based business integration solutions. As noted above, Curry also fails to teach a computer implementation of Curry’s method into which Huang’s “design patterns” might be combined. The proposed combination of Huang with Curry would not result in

program instructions executable by a processor to apply one or more design patterns to the integrated Web Service architecture, as recited in Applicant's claim 18.

In further regard to claim 18, the Office has failed to establish a proper *prima facie* reason to combine the references.

The Office Action asserts that "it would have been obvious...to have modified Curry such that the web service architecture is generated in accordance with one or more integration design patterns as taught by Huang because design patterns are well known in the art and commonly used by programmers since design patterns provide the benefit of capturing a standard solution to a common programming problem for reuse." However, Curry teaches four specific design patterns – a Composite pattern, a Mediator pattern, a Command pattern, and a State pattern. It is unclear as to how Curry's system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry's "Web services development method" as disclosed. Curry and Huang each propose distinctly different methods for achieving similar results. Modifying Curry with Huang's specific "design patterns" would appear to necessitate significant changes in Curry's system. "If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious." *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). Thus, one of ordinary skill would not have combined the teachings of Huang with the teachings of Curry in the manner proposed by the Office. Accordingly, the Office has failed to establish a *prima facie* case of obviousness.

In further regard to claim 18, the cited references fail to teach at least the features of, program instructions executable by a processor to provide output indicating the generated integrated Web Service architecture for implementing the specific integrated Web Service business system.

As previously mentioned, Curry does not teach program instructions executable

by a processor to implement the methods illustrated in Curry's Figures. Moreover, Curry does not teach program instructions executable by a processor to provide output indicating a generated integrated Web Service architecture for implementing a specific integrated Web Service business system. Huang only conceptually describes a framework, and does not teach or even suggest a computer implementation of the framework, nor does Huang teach or suggest program instructions executable by a processor to provide output indicating a generated integrated Web Service architecture for implementing a specific integrated Web Service business system. Moreover, no combination of Huang and Curry teaches these features as recited in Applicant's claim 18.

Thus, for at least the reasons presented above, the rejection of claim 18 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks as those above regarding claim 18 also apply to claims 58 and 82.

In regard to claim 31, as a first matter, neither of the cited references (Curry and Huang) teach computer-implemented methods for generating integrated Web Service architectures.

As noted above in regards to claim 1, nowhere does Curry teach that the method illustrated in FIG. 1 is a computer-implemented method, and Curry does not describe a computer system that implements the method illustrated in FIG. 1. At most, Curry describes that particular steps or sub-steps of Curry's method may be assisted by or performed using existing software tools. Huang only conceptually describes a framework for developing Web services-based business integration solutions, and does not teach or even suggest a computer implementation of the framework for developing Web services-based business integration solutions.

In contrast to both Curry and Huang, claim 31 recites an integrated Web Service architecture for an integrated Web Services business system that is generated by a computer-implemented integrated Web Services architecture design service according to

a vendor-independent architecture framework for integrating Web Services technologies with business systems comprising a plurality of heterogeneous components in accordance with a structured integration methodology and one or more design patterns.

Furthermore, as noted above in regard to claim 1, in paragraph [0055] Curry teaches “The framework includes a pre-built architecture that allows developers to rapidly create applications based on business components and web services.” **Curry does not teach an integrated Web Service architecture for an integrated Web Services business system that is generated by an integrated Web Services architecture design mechanism implemented on one or more computers.** Instead, Curry clearly teaches a “standard framework architecture” that clearly pre-exists and that may allegedly be used to “rapidly create applications based on business components and web services.”

Neither Curry nor Huang, alone or in combination, teach a system as recited in Applicant’s claim 31 when the claim is viewed as a whole.

In further regard to claim 31, the cited references fail to teach at least the features of, an integrated Web Services business system comprising a plurality of tiers implemented according to an integrated Web Service architecture for the integrated Web Services business system, wherein the plurality of integration tiers comprises a client tier, a presentation tier, a business tier, an integration tier, and a resources tier, as recited in amended claim 31.

Neither Curry nor Huang, alone or in combination, teach the above limitations as recited in claim 1. Curry teaches a “framework structure” that includes “includes primarily three main sub-architectures” (paragraphs [0056]-[0059]). Curry, however, does not teach an integrated Web Services business system comprising a plurality of integration tiers including a client tier, a presentation tier, a business tier, an integration tier, and a resources tier. Likewise, Huang does not teach an integrated Web Services

business system comprising a plurality of integration tiers including a client tier, a presentation tier, a business tier, an integration tier, and a resources tier.

In further regard to claim 31, the Office has failed to establish that the cited references teach an integrated Web Service architecture for an integrated Web Services business system that is generated by an integrated Web Services architecture design mechanism implemented on one or more computers according to a vendor-independent architecture framework for integrating Web Services technologies with business systems comprising a plurality of heterogeneous components in accordance with a structured integration methodology and one or more design patterns.

The Office Action admits that Curry does not teach that the “integrated Web Services business system is configured according to one or more design patterns,” and relies on Huang to teach these features, citing page 18, second column, paragraph 2 and page 20, first column paragraphs 2 and 3 and second column, paragraphs 1-3. In Section 3, at page 18, second column, first full paragraph, Huang teaches “In this section, we propose a framework to develop Web services-based business integration solutions.” In the first paragraph cited from page 20, Huang teaches “In essence, this framework uses four different design patterns...” However, Huang only conceptually describes this framework, and does not teach or even suggest a computer implementation of the framework for developing Web services-based business integration solutions. As noted above, Curry also fails to teach a computer implementation of Curry’s method into which Huang’s “design patterns” might be combined. The proposed combination of Huang with Curry would not result in an integrated Web Service architecture for an integrated Web Services business system that is generated by an integrated Web Services architecture design mechanism implemented on one or more computers according to a vendor-independent architecture framework for integrating Web Services technologies with business systems comprising a plurality of heterogeneous components in accordance with a structured integration methodology and one or more design patterns, as recited in Applicant’s claim 31.

In further regard to claim 31, the Office has failed to establish a proper *prima facie* reason to combine the references.

The Office Action asserts that “it would have been obvious...to have modified Curry such that the web service architecture is generated in accordance with one or more integration design patterns as taught by Huang because design patterns are well known in the art and commonly used by programmers since design patterns provide the benefit of capturing a standard solution to a common programming problem for reuse.” However, Curry teaches four specific design patterns – a Composite pattern, a Mediator pattern, a Command pattern, and a State pattern. It is unclear as to how Curry’s system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry’s “Web services development method” as disclosed. Curry and Huang each propose distinctly different methods for achieving similar results. Modifying Curry with Huang’s specific “design patterns” would appear to necessitate significant changes in Curry’s system. “If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious.” *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). Thus, one of ordinary skill would not have combined the teachings of Huang with the teachings of Curry in the manner proposed by the Office. Accordingly, the Office has failed to establish a *prima facie* case of obviousness.

Thus, for at least the reasons presented above, the rejection of claim 31 is not supported by the cited art and removal thereof is respectfully requested.

In regard to claim 42, as a first matter, neither of the cited references (Curry and Huang) teach computer-implemented means for generating integrated Web Service architectures.

As noted above in regards to claim 1, nowhere does Curry teach that the method illustrated in FIG. 1 is a computer-implemented method, and Curry does not describe a computer system that implements the method illustrated in FIG. 1. At most, Curry describes that particular steps or sub-steps of Curry's method may be assisted by or performed using existing software tools. Huang only conceptually describes a framework for developing Web services-based business integration solutions, and does not teach or even suggest a computer implementation of the framework for developing Web services-based business integration solutions.

In contrast to both Curry and Huang, claim 42 recites computer-implemented means for generating an integrated Web Services architecture for integrating a specific Web Service with a specific business system.

Furthermore, as noted above in regard to claim 1, in paragraph [0055] Curry teaches "The framework includes a pre-built architecture that allows developers to rapidly create applications based on business components and web services." **Curry does not teach an integrated Web Service architecture that is generated by means for generating, on one or more computers, integrated Web Services architectures.** Instead, Curry clearly teaches a "standard framework architecture" that clearly pre-exists and that may allegedly be used to "rapidly create applications based on business components and web services."

Neither Curry nor Huang, alone or in combination, teach a system for integrating Web Services with business systems as recited in Applicant's claim 42 when the claim is viewed as a whole.

In further regard to claim 42, the cited references fail to teach at least the features of computer-implemented means for applying a Web Services structured methodology and one or more design patterns to the generated integrated Web Service architecture to identify heterogeneous components for the integrated Web

Service architecture and to organize the heterogeneous components according to the integrated Web Service architecture, as recited in amended claim 42.

The Office Action relies on Curry to teach means for applying a Web Services structured methodology to the integrated Web Service architecture, citing paragraphs [0055]-[0059]. However, the Office Action asserts that the “pre-built architecture” in paragraph [0055] teaches the integrated Web Service architecture, and thus equates Curry’s “framework structure” or “framework architecture” with claim 42’s integrated Web Service architecture. Paragraph [0055] states that “A framework review 18 can then be performed on the interface templates.” Thus, step 18 of Curry’s FIG. 1 is performed on the “interface templates,” using the “framework architecture”, and is not performed on Curry’s “pre-built architecture” which the Office equates with claim 42’s integrated Web Service architecture.

Moreover, paragraph [0060] of Curry states that “During the framework review 18 of the interface templates, a developer reviews the prototype from a framework perspective and insures that the interface templates are achievable in a framework context.” Thus, not only is step 18 performed on Curry’s “interface templates” and not on Curry’s framework architecture, but the framework review step is performed by a human “developer” and not by “computer-implemented means.”

Furthermore, amended claim 42 recites “computer-implemented means for applying a Web Services structured methodology [and one or more design patterns] to the generated integrated Web Service architecture to identify heterogeneous components for the integrated Web Service architecture and to organize the heterogeneous components according to the integrated Web Service architecture.” Applicant can find nothing in Curry or Huang, alone or in combination, that teaches this subject matter as recited in claim 42.

In further regard to claim 42, the cited references fail to teach at least the features of, wherein said computer-implemented means for applying a Web Services

structured methodology and one or more design patterns to the generated integrated Web Service architecture comprises means for providing integration and interoperability with the integrated Web Service architecture for existing business functionality of the specific business system, as recited in amended claim 42.

Applicant can find nothing in Curry or Huang, alone or in combination, that teaches this subject matter as recited in amended claim 42.

In further regard to claim 42, the cited references fail to teach at least the features of, computer-implemented means for providing output indicating a generated integrated Web Service architecture for integrating a Web Service with a business system.

Curry does not teach computer-implemented means for providing output indicating a generated integrated Web Service architecture for integrating a Web Service with a business system. Huang only conceptually describes a framework, and does not teach or suggest computer-implemented means for providing output indicating a generated integrated Web Service architecture for integrating a Web Service with a business system. Moreover, no combination of Huang and Curry teaches these features as recited in Applicant's claim 42.

In further regard to claim 42, the Office has failed to establish that the cited references teach computer-implemented means for applying a Web Services structured methodology and one or more design patterns to the generated integrated Web Service architecture to identify heterogeneous components for the integrated Web Service architecture and to organize the heterogeneous components according to the integrated Web Service architecture.

The Office Action admits that Curry does not teach that the "integrated Web Services business system is configured according to one or more design patterns," and relies on Huang to teach these features, citing page 18, second column, paragraph 2 and

page 20, first column paragraphs 2 and 3 and second column, paragraphs 1-3. In Section 3, at page 18, second column, first full paragraph, Huang teaches “In this section, we propose a framework to develop Web services-based business integration solutions.” In the first paragraph cited from page 20, Huang teaches “In essence, this framework uses four different design patterns...” However, Huang only conceptually describes this framework, and does not teach or even suggest a computer implementation of the framework for developing Web services-based business integration solutions. As noted above, Curry also fails to teach a computer implementation of Curry’s method into which Huang’s “design patterns” might be combined. The proposed combination of Huang with Curry would not result in computer-implemented means for applying a Web Services structured methodology and one or more design patterns to the generated integrated Web Service architecture to identify heterogeneous components for the integrated Web Service architecture and to organize the heterogeneous components according to the integrated Web Service architecture, as recited in Applicant’s claim 42.

In further regard to claim 42, the Office has failed to establish a proper *prima facie* reason to combine the references.

The Office Action asserts that “it would have been obvious...to have modified Curry such that the web service architecture is generated in accordance with one or more integration design patterns as taught by Huang because design patterns are well known in the art and commonly used by programmers since design patterns provide the benefit of capturing a standard solution to a common programming problem for reuse.” However, Curry teaches four specific design patterns – a Composite pattern, a Mediator pattern, a Command pattern, and a State pattern. It is unclear as to how Curry’s system would be modified with these specific design patterns, much less how the modification could be made without changing the principle of operation of Curry’s “Web services development method” as disclosed. Curry and Huang each propose distinctly different methods for achieving similar results. Modifying Curry with Huang’s specific “design patterns” would appear to necessitate significant changes in Curry’s system. “If the proposed modification or combination of the prior art would change the principle of operation of

the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious.” *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). Thus, one of ordinary skill would not have combined the teachings of Huang with the teachings of Curry in the manner proposed by the Office. Accordingly, the Office has failed to establish a *prima facie* case of obviousness.

Thus, for at least the reasons presented above, the rejection of claim 42 is not supported by the cited art and removal thereof is respectfully requested.

Applicant also asserts that the rejection of numerous ones of the dependent claims under 35 U.S.C. § 103(a) is further unsupported by the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

The Office Action rejected claims 4, 26, 32, 48, 66, 72 and 90 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry in view of Huang and further in view of Connell, et al. (U.S. Publication 2003/0074401) (hereinafter “Connell”). However, since the rejection has been shown to be unsupported for the independent claims from which these claims depend, a further discussion of this rejection is not necessary at this time.

The Office Action rejected claims 7, 11, 12, 25, 36, 40, 44, 51, 55, 65, 75, 79 and 89 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Curry in view of Huang, and further in view of Chappell, et al. (“Java Web Services”) (hereinafter “Chappell”). However, since the rejection has been shown to be unsupported for the independent claims from which these claims depend, a further discussion of this rejection is not necessary at this time.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-66303/RCK.

Respectfully submitted,

/Robert C. Kowert/

Robert C. Kowert, Reg. #39,255
Attorney for Applicant

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: March 5, 2010